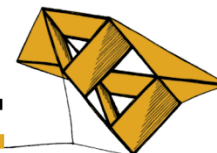




STUDENTSKÁ VĚDECKÁ KONFERENCE 2014



Automatické sledování obličeje

Martin Volavka¹

1 Úvod

Hlavním cílem úlohy automatického sledování obličeje je pomocí obyčejné komerční webkamery připevněné na pohyblivém stojanu snímat obraz aktuální scény, vyhledat v ní lidský obličej a podle jeho pozice otočit kameru tak, aby byl obličej v centru snímané scény.

Jedná se o kompletní ucelený řídicí systém, jehož vstupem je detekovaná poloha obličeje na aktuálním snímku, výstupem je nastavení horizontální a vertikální polohy kamery. Poloha kamery je měněna pomocí dvou modelářských servopohonů řízených vývojovým kitem Arduino UNO. Toto zařízení komunikuje prostřednictvím sériové linky s běžícím algoritmem v programovém prostředí Python.

Jelikož běžící program v Pythonu pracuje s digitalizovaným obrazem v reálném čase, je nutné úlohu omezit určitými provozními podmínkami.

- Snímání scény je prováděno při klasickém nebo umělém osvětlení (zářivka, žárovka).
- Kameru lze otáčet v horizontálním a vertikálním směru v rozmezí $\pm 90^\circ$.
- Pro účely sledování bude využit pouze jeden, nejbližší detekovaný lidský obličej vzhledem ke snímací kameře.
- Pokud nedojde k detekci obličeje, kamera zůstane na stávající pozici.
- Při menší vzdálenosti (pozorovatel – kamera), tj. do 1.5 m, lze detekovat polohu očí uživatele.
- Změna jednotlivých poloh servopohonů bude prováděna postupně, tj. s krokem jednoho stupně, aby nedocházelo k rozkmitání systému.

Pro úlohu detekce obličeje byla použita již vymyšlená metoda Haarova kaskádního klasifikátoru. Metoda dokáže rozlišit (detekovat) obličej s velikou přesností, ale pouze v případě, kdy je obličej situován na snímku ve svislém směru. Toleranční pásmo svislého směru je zhruba $\pm 20^\circ$ vzhledem k vertikální ose obličeje. Pro rychlou a úspěšnou detekci je důležité klasifikátor správně nastavit, tj. zvolit správné hodnoty parametrů.

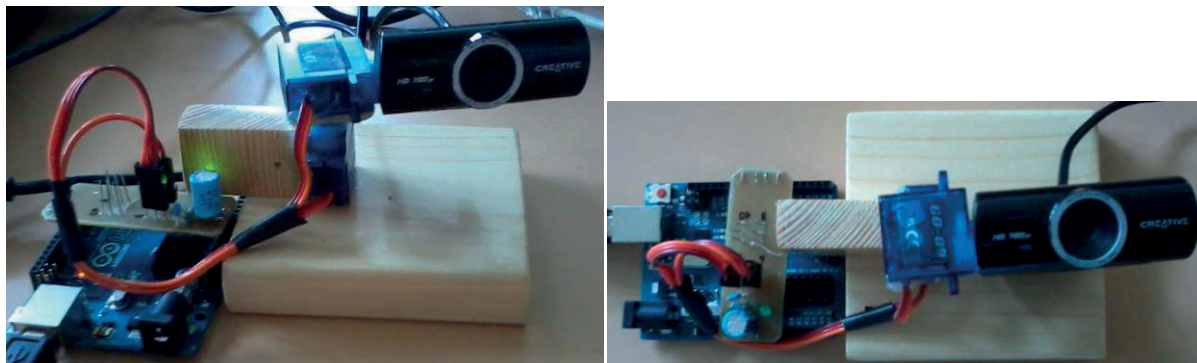
Grafické znázornění detekce obličeje je provedeno pomocí modrého vykresleného obdélníku, jehož středem je obličej a jehož šířka odpovídá odhadované velikosti (kompromis mezi šířkou a délkou) obličeje. Střed snímku a střed obličeje jsou znázorněny pomocí červených obdélníků. Pokud dojde k detekci očí, bude jejich umístění na snímku zobrazeno pomocí zelených obdélníků. Algoritmus také poskytuje informaci o úhlu natočení spojnice detekovaných očí vzhledem k souřadnicové soustavě pořízeného snímku. Tato spojnice je vykreslena červenou čarou.

Během detekce očí dochází vlivem nerovnoměrného osvětlení k detekci jiných částí obličeje, především úst a nosních dírek. Proto je důležité, aby byl pozorovatel co nejrovnoměrněji osvětlen a přímo hleděl do kamery. Oči lze detekovat i v případě nosí-li pozorovatel dioptrické brýle s průhlednými skly a slabými obroučky.

¹ student navazujícího studijního programu Inženýrská informatika, obor Řídicí a rozhodovací systémy, specializace Automatické řízení, e-mail: martinvolavka@seznam.cz

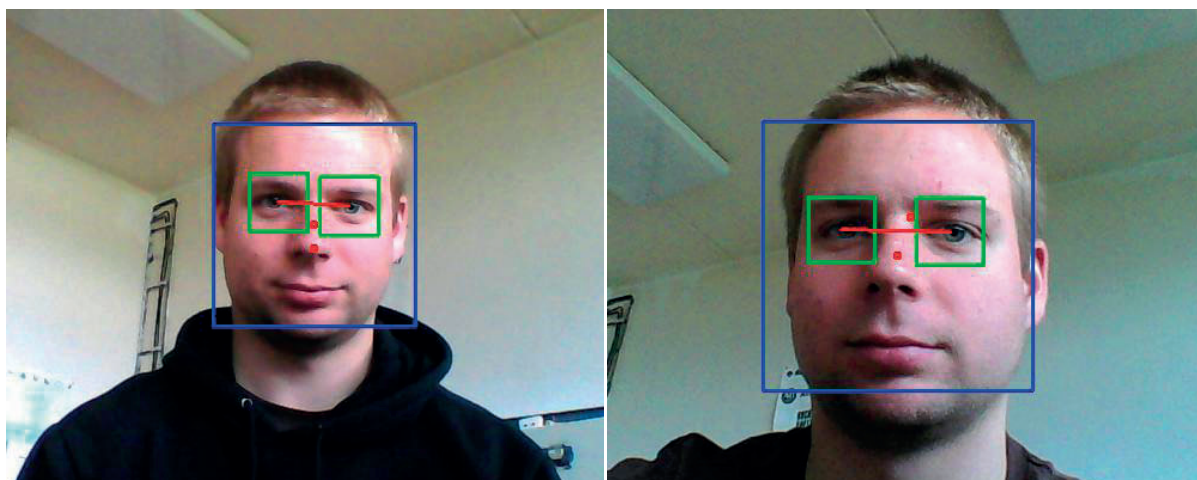
2 Konstrukční zpracování systému a grafické znázornění výstupu

Konstrukce celého systému je velmi jednoduchá (viz Obrázek 1.), obsahuje dřevěný stojan vybaven dvěma servopohony, na druhém (horizontálním) pohonu je umístěna kamera, řídicí člen Arduino UNO a vodiče umožňující komunikaci s běžným algoritmem v počítači.



Obrázek 1: Konstrukční provedení systému

Výstupem systému je, jak bylo řečeno, upravená poloha kamery tak, aby se střed snímaného obrazu nacházel v blízkosti (tj. ± 40 obrazových bodů) detekovaného obličeje. Tato situace je vyobrazena na Obrázku 2.



Obrázek 2: Výstup systému

Algoritmus automatického sledování lidského obličeje funguje i pro větší vzdálenosti pozorovatel – kamera. Tato vzdálenost byla experimentálně určena a odpovídá cca 8 metrům při použití standardní webkamery.

Literatura

OpenCV 3.0.0-dev documentation. *Face Detection using Haar Cascades*.

Dostupné z:

http://docs.opencv.org/trunk/doc/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html

Arduino Uno. *Communication and programming*.

Dostupné z:

<http://arduino.cc/en/Main/arduinoBoardUno>